# Machine Learning Week 4. Neural Networks.

## 1. Non-linear hypothesis.

- non-linear classification. → a lot of terms.
  - not scalable, $O(n^2)$, $O(n^3)$

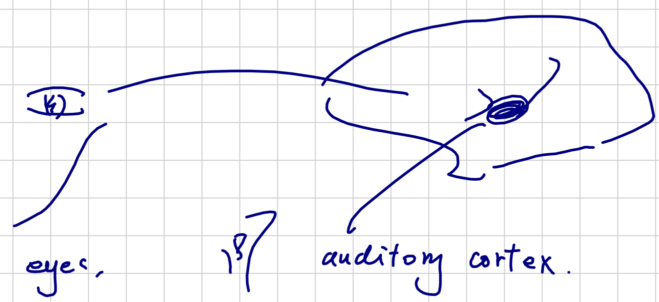
- application: computer vision
  - labelled training set ⇒ train model.

- neural network is useful for non-linear, large scale classification

## 2. Neural networks

- origin → "neuromorphic" to mimic how the brain works.
- recent resurgence: computer (hardware) advancement.

- The "one learning algorithm" hypothesis



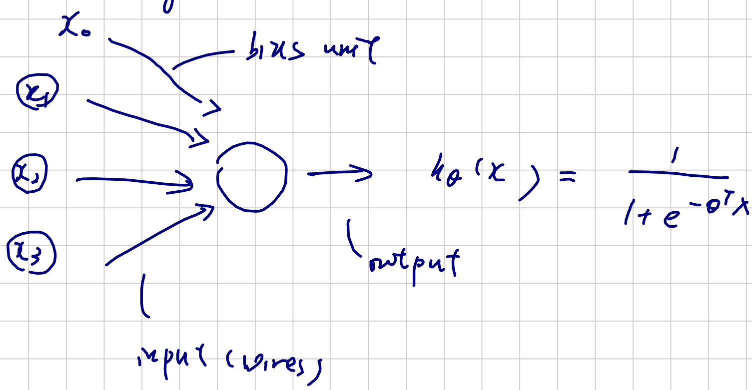
if we re-wire to eyes, auditory cortex.
the auditory will learn how to "see".

"neural re-wiring" → there exists "one algorithm" that can do generalized learning.

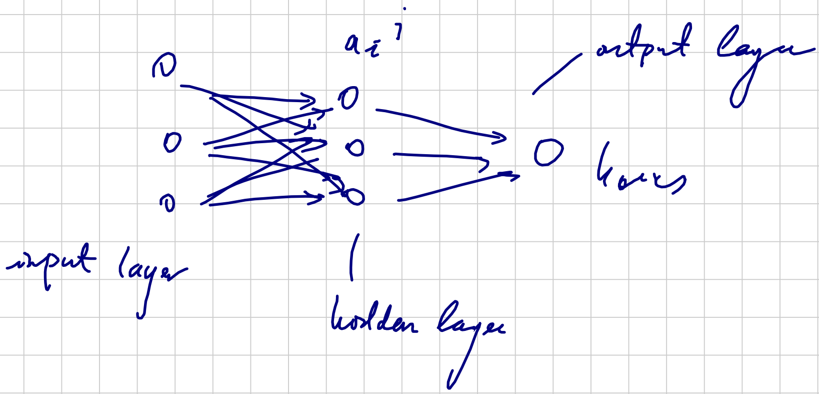
3. Neural network modal representation

- interneuron communication - sends electric pulses

- Neuron model: logistic unit




sigmoid (logistic) activation function



4. Model Representation II.

Forward Propagation $\longrightarrow$ vectorized implementation.

$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$

$$\Rightarrow a_1^{(2)} = z_2^{(2)}$$

$$x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad z^{(2)} = \begin{pmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{pmatrix} \left.\begin{array}{l} = \Theta^{(1)} (x) \\ = \Theta^{(1)} \cdot a^{(1)} \end{array}\right\} \text{define}$$

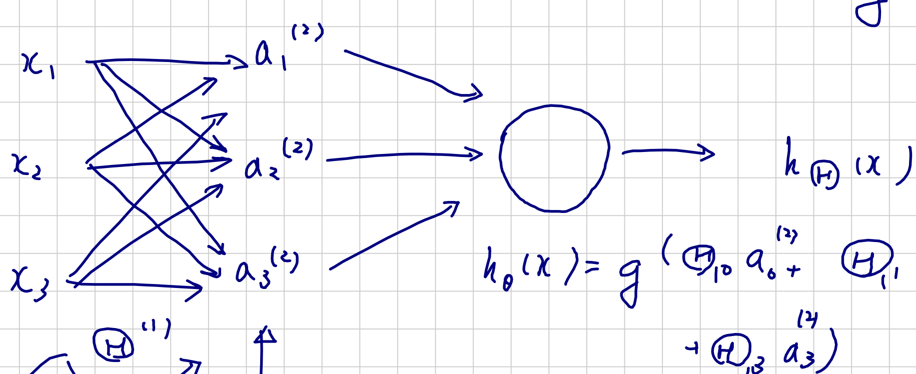
$$\underset{\mathbb{R}^3}{a^{(2)}} = g(\underset{\mathbb{R}^3}{z^{(2)}})$$

element wise.

Add. $a_0^{(2)} = 1$, $a^{(2)} \in \mathbb{R}^{(4)}$

$z^{(3)} = \Theta^{(2)} a^{(2)}$, $h_\Theta(x) = a^{(3)} = g(z^3)$

Neural network learning its own features : flexibility to learn features first [hidden layer] before feeding into logistic regression




$$h_\theta(x) = g(\Theta_{10} a_0^{(2)} + \Theta_{11} a_1^{(2)} + \Theta_{12} a_2^{(2)} + \Theta_{13} a_3^{(2)})$$

mapping function (original $\rightarrow$ process) features.

"processed" features that feed into the original logistic regression.

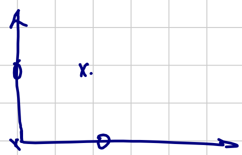
standard logistic model.

Other Network Architecture.

- multiple hidden layers $\longrightarrow$ complexity, non-linearity $\uparrow$

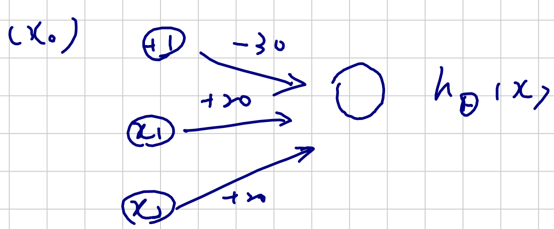
## 6. Examples and Intuition (1)

Non-linear classification example. XOR / XNOR.

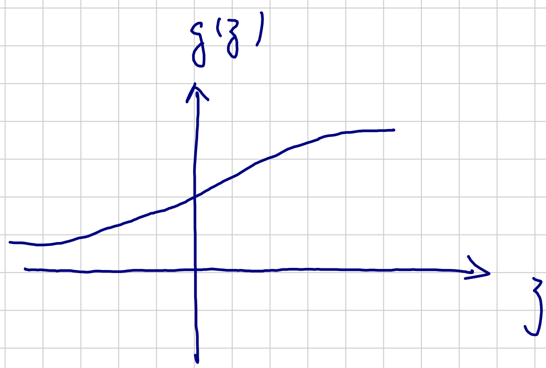



$y = x_1 \text{ XOR } x_2$

simple example. $x_1, x_2 \in \{0, 1\}$.




$$h_\Theta(x) = g(\underbrace{-30}_{\Theta_{10}^{(1)}} \cdot 1 + \underbrace{20}_{\Theta_{11}^{(1)}} \cdot x_1 + \underbrace{20}_{\Theta_{12}^{(1)}} x_2)$$




$x_1$ & $x_2$

| $x_1$ | $x_2$ | $h_\Theta(x)$ |
|---|---|---|
| 0 | 0 | $g(-30) = 0$ |
| 0 | 1 | $g(-10) = 0$ |
| 1 | 0 | $g(-10) = 0$ |
| 1 | 1 | $g(10) = 1$ |

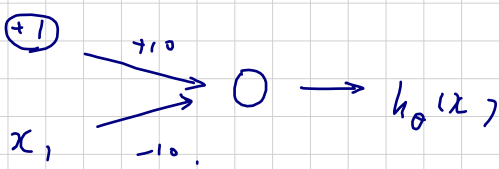
consider. $\Theta^{(1)} = \begin{pmatrix} -10 \\ 20 \\ 20 \end{pmatrix}$

| $x_1$ | $x_2$ | $z$ | $g(z)$ |
|---|---|---|---|
| 0 | 0 | -10 | 0 |
| 0 | 1 | 10 | 1 |
| 1 | 0 | 10 | 1 |
| 1 | 1 | 30 | 1 |

$x_1$ or $x_2$

6. Example and Intuition. (II)

- Not gate.



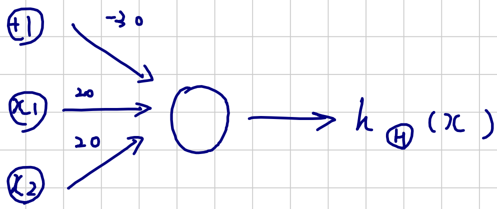

use large negative weight here to get NOT gate

$h_\Theta(x) = g(10 - 20x_1)$

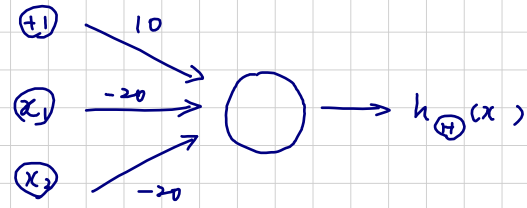
| $x_1$ | $z$ | $g$ |
|---|---|---|
| 0 | 10. | 1 |
| 1 | -10 | 1 |

- ex. (NOT. $x_1$) AND. (NOT $x_2$) $=1 \iff x_1 = 0,\ x_2 = 0$.

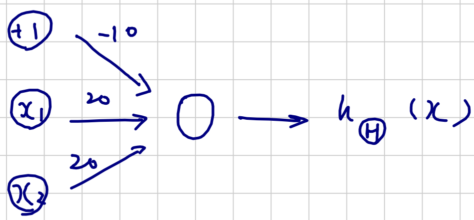
Summary.



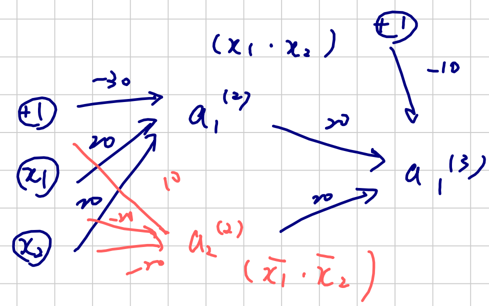

$x_1$ AND $x_2$




$\bar{x}_1$ AND $\bar{x}_2$




$x_1$ OR $x_2$




$x_1$ XNOR $x_2$

$$\overline{x_1 \oplus x_2} = x_1 x_2 + \bar{x}_1 \cdot \bar{x}_2$$

7. Multiclass classification.

- multiple output units. (each unit classifies true/false on each bin)